

# Compiler Lec

~~Strings~~

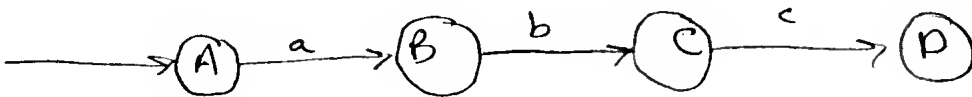
Epsilon move

— يمكن نقله A إلى c ~ ~ ~ (Consuming) للمبار.

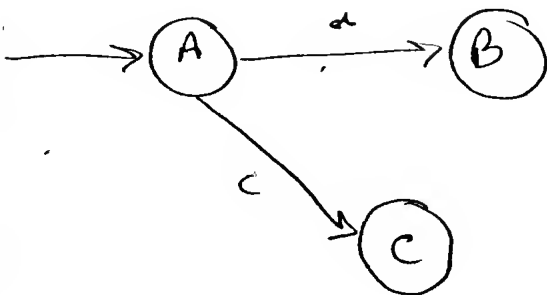
— أعطى اختياراً كثيرة.

— يمكن تحريكه (input) و (input string) يساراً  
مختلفة (كل الحروف تؤدي إلى ردود).

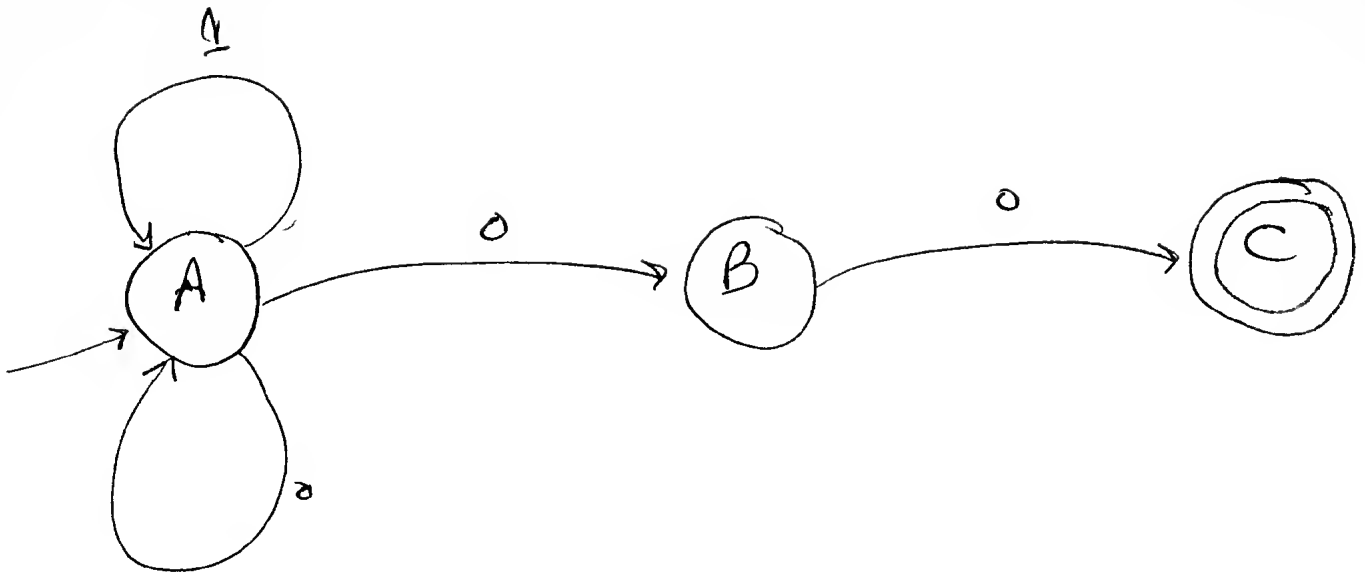
abc



Deterministic  
Finite  
Automata.



Non Deterministic  
Finite  
Automata.



→ Non Deterministic .  
 ← عساً بعد ( 2 transition ) لا 0 1 لبداية .

Note Input : 1 0 0  
 state : ~~{A}~~ {A} {A,B} {A,B,c}

نضعها بين قوسين هكذا  
 multiple choises بسبب

← لو آخر state فيها (accept state) اذا

الحالة دي Accepted .

Rejected

← لو وصلت لنهاية مصفوفة (Accept state)

← لو لم اقبل حتى نهاية ال string (state) حتى لو (Accepted)

RE ← يوافق على طريقه ال (operation)

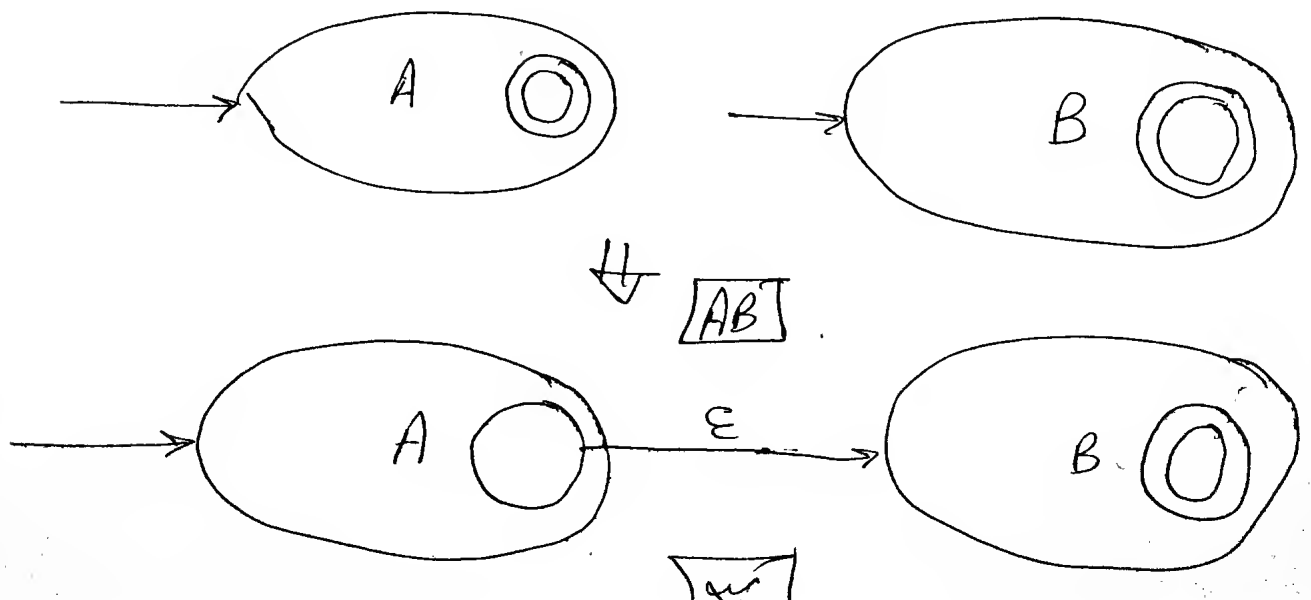
~~more~~ No choices ← DFA (states) ل (diagrams) يرمز

" " " ← NDFA  
(more ~~the~~ choices to consider)

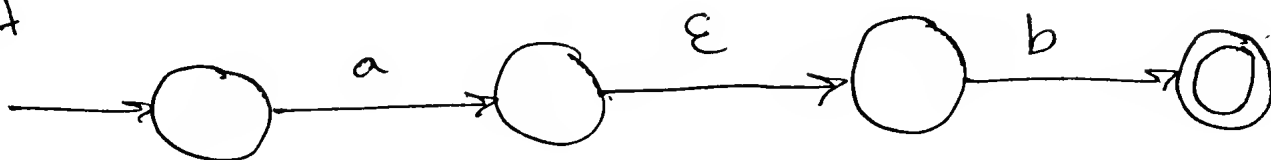
← اللغة- يوافقوا نفس ال (formal language)

DFA ← المسار واحد لا يتغير .

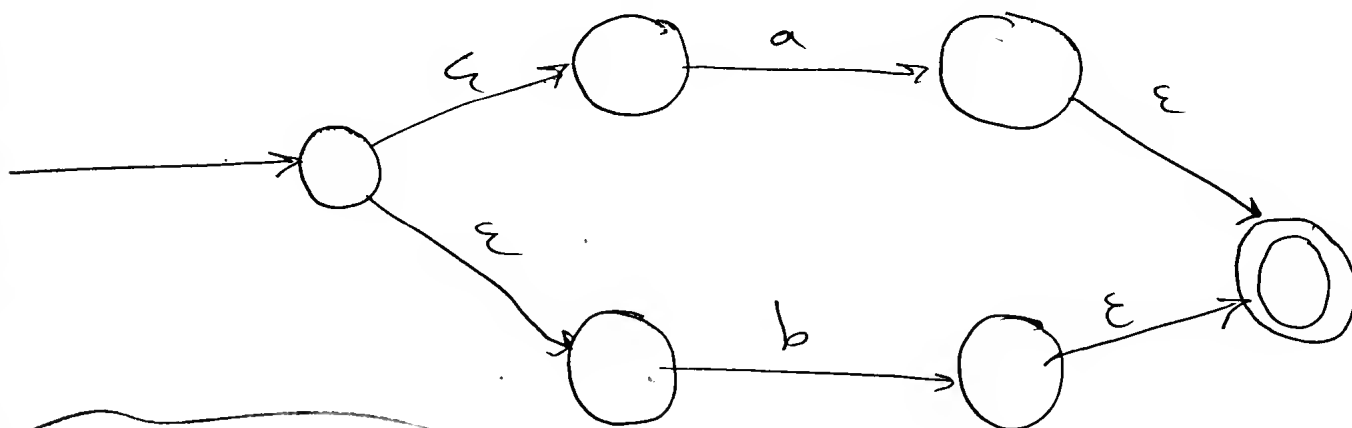
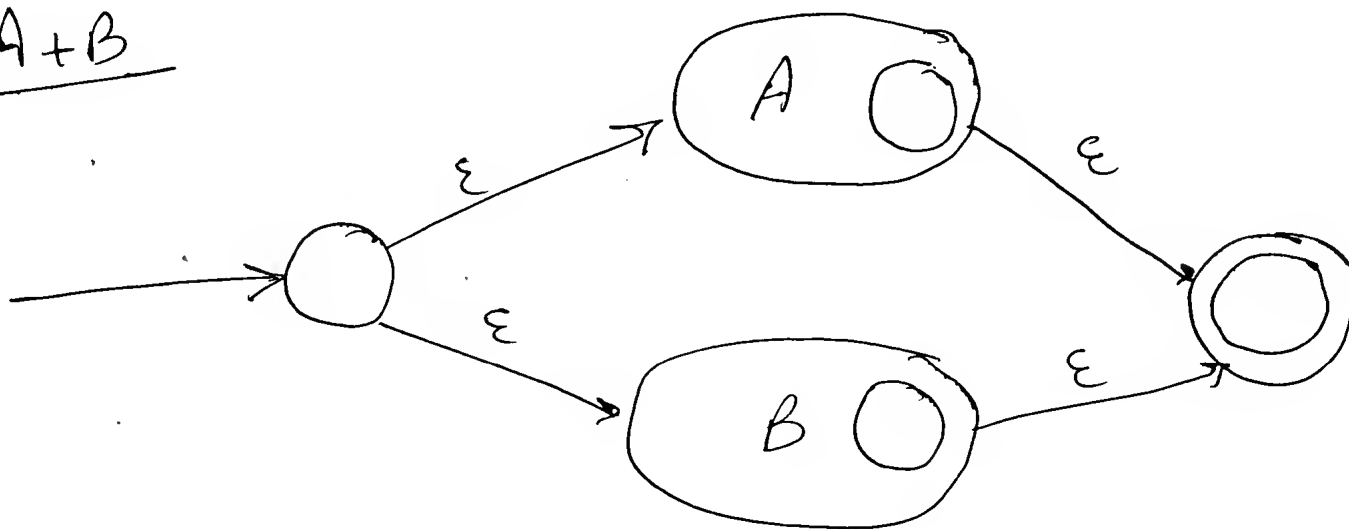
RE مختلفة ممكنة اعمل فيها DFA مختلفة بس  
لو عملت (analyze) ليهم هيكملعولي نفس النتيجة .  
لكر لو عملت (NDFA) ~~هيكملعولي~~ هيبقى الحل واحد .  
'A' 'B'



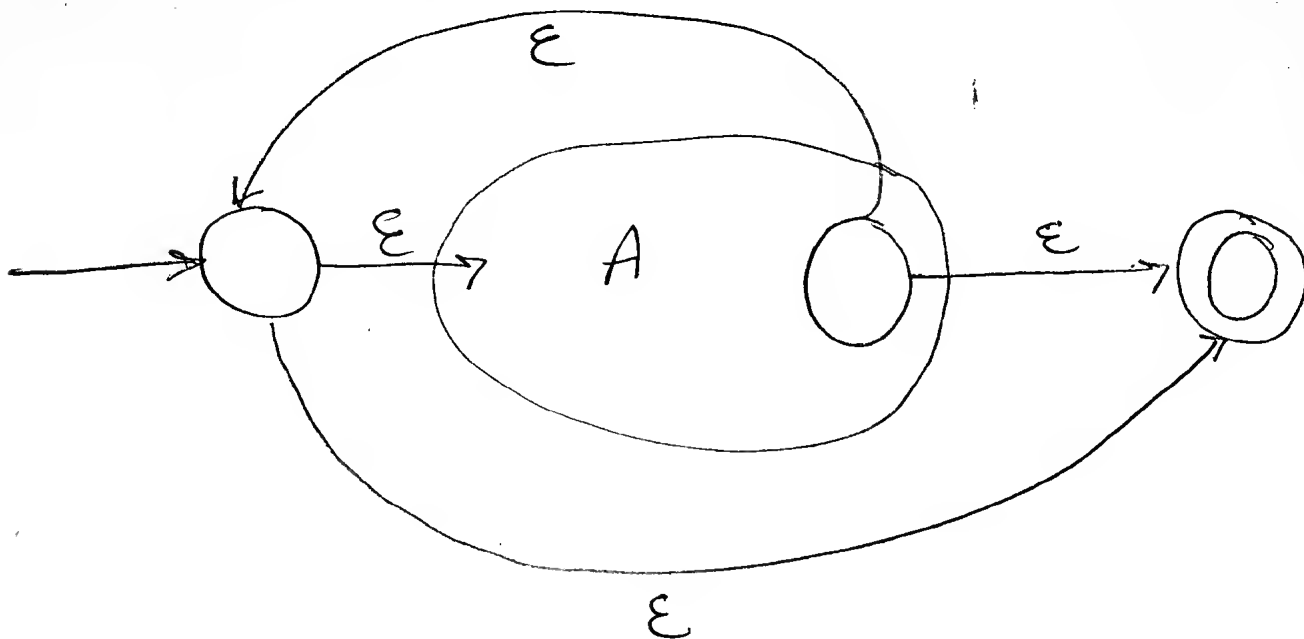
AB



A+B

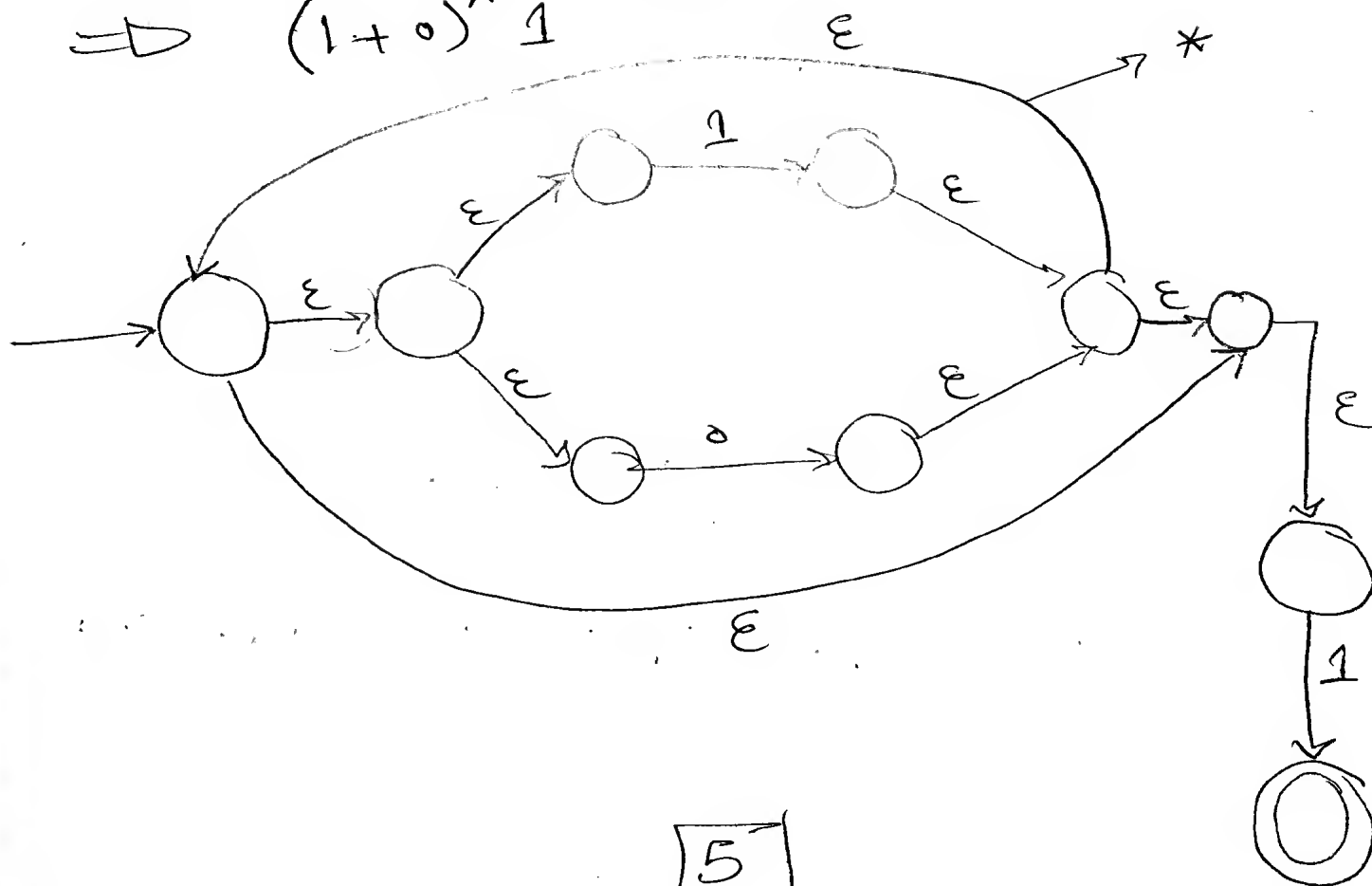


9



→ RE to NFA (ε)

⇒  $(1+0)^* 1$



5

~~A~~

Quiz

رقم B

0+1\* سے تقریباً وحاید ای عدد صدمراہ  
بالما تقریباً ہنر .

$\epsilon$ -closure  $\{S\}$

Function (Function) ال (argument) بتاھا (state)

حل ال (states) الی قدر توہل ال (state)  $[S]$  صر عر ما نقل Consuming

رسة هفده  
31

$$\epsilon\text{-closure } \{B\} = \{B, C, D\}$$

$$\epsilon\text{-closure } \{G\} = \{G, H, I, A, B, C, D\}$$

input character ينقل (Set of next states) multiple choices

6

$a(X)$   $\rightarrow$  set of states

$$\epsilon\text{-closure}\{G\} = \{G, H, A, I, A, B, C, D\}$$

$\downarrow$   
 $X$

وكل حرف من غير  $X$  (small character) .  $y$  or  $x$  ليس

$$a(X) = \{y \mid x \in X_n \text{ } x \xrightarrow{a} y\}$$

$$= Y$$

$\epsilon\text{-closure}\{Y\}$   $\leftarrow Y$  (نفسه)  $\rightarrow$   $Y$  (نفسه)

start state  $\rightarrow \epsilon\text{-closure}$  (a

(data)  $\rightarrow$  input ch.  $\rightarrow$  (consuming)  $\rightarrow$   $b$

$c$

$d$

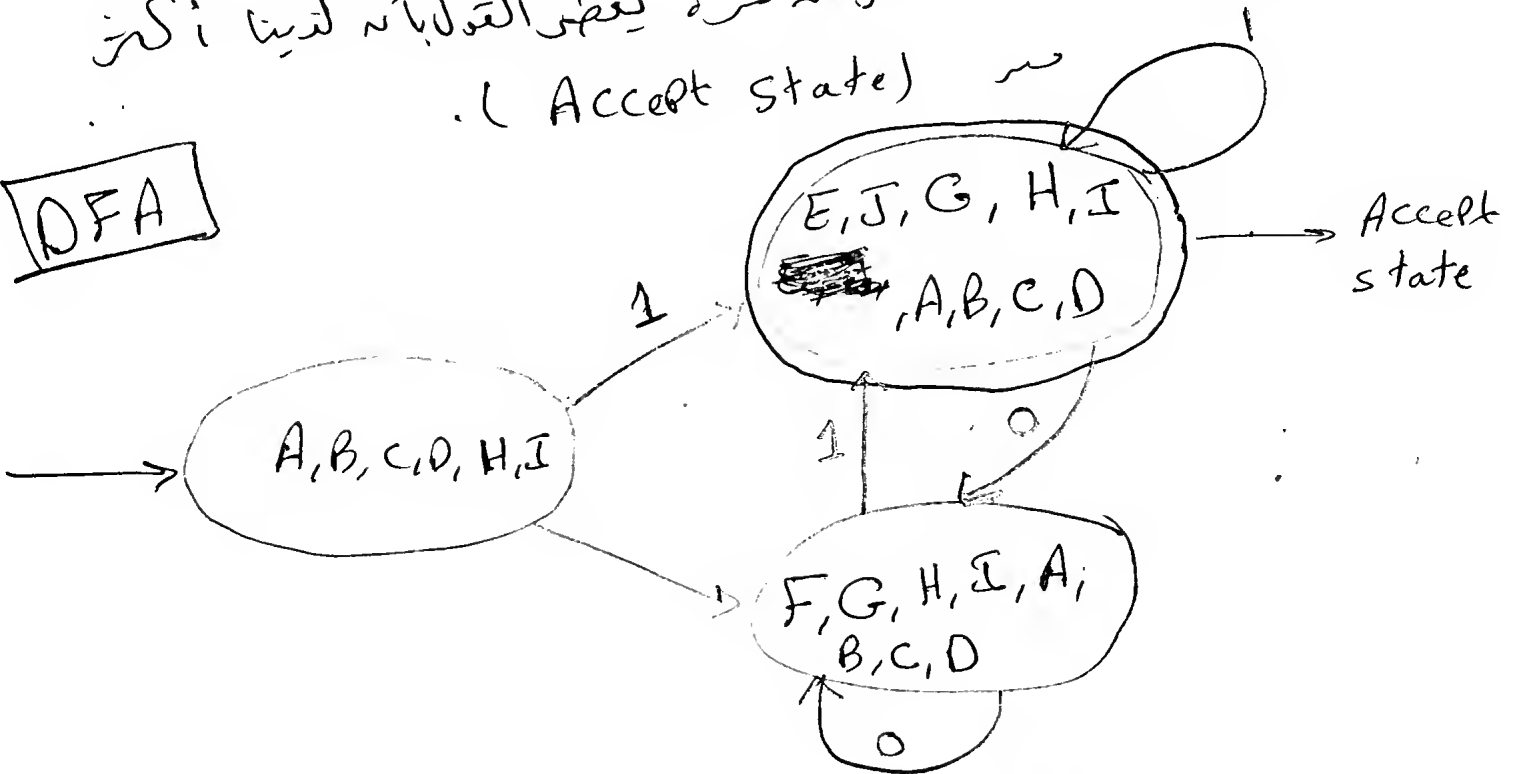
1)

$$\epsilon\text{-closure}\{A\} = \{A, B, C, D, H, I\}$$

أما مثل فنظروا غير هنا .

لأنه نظروا أكثر مرة يعطى التوليد لدينا أكثر . (Accept state)

DFA



$$\textcircled{1}(X) = \{E, J\}$$

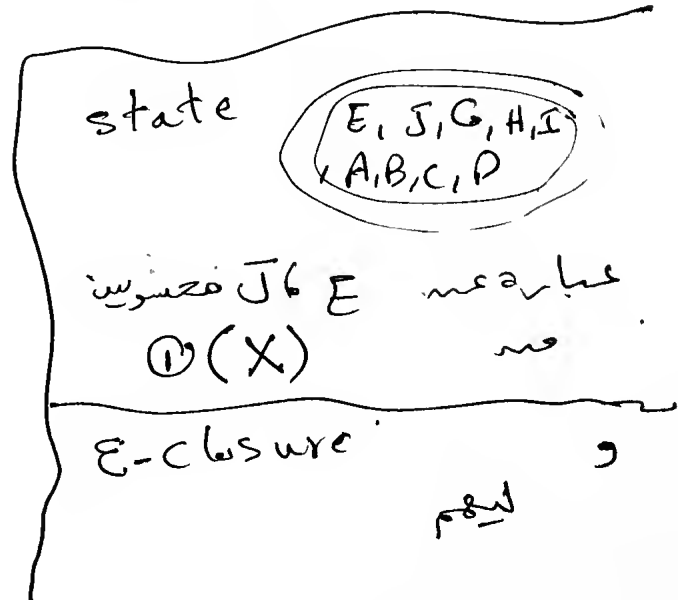
$$C \xrightarrow{1} E$$

$$I \xrightarrow{1} J$$

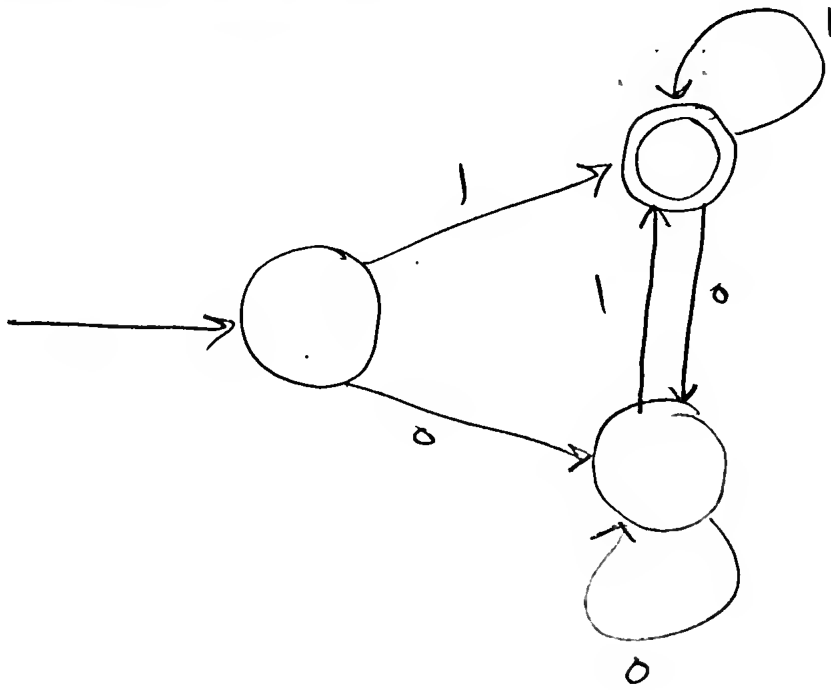
$$\epsilon\text{-closure}\{\textcircled{1}(X)\}$$

$$\epsilon\text{-closure}\{E, J\}$$

⊆



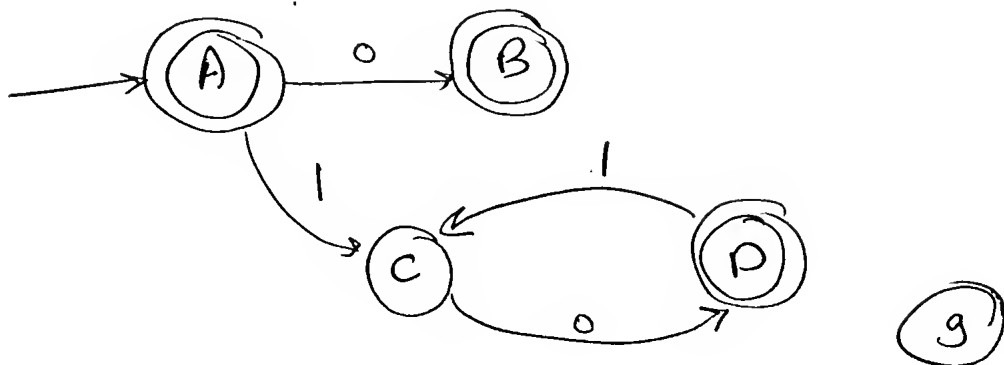
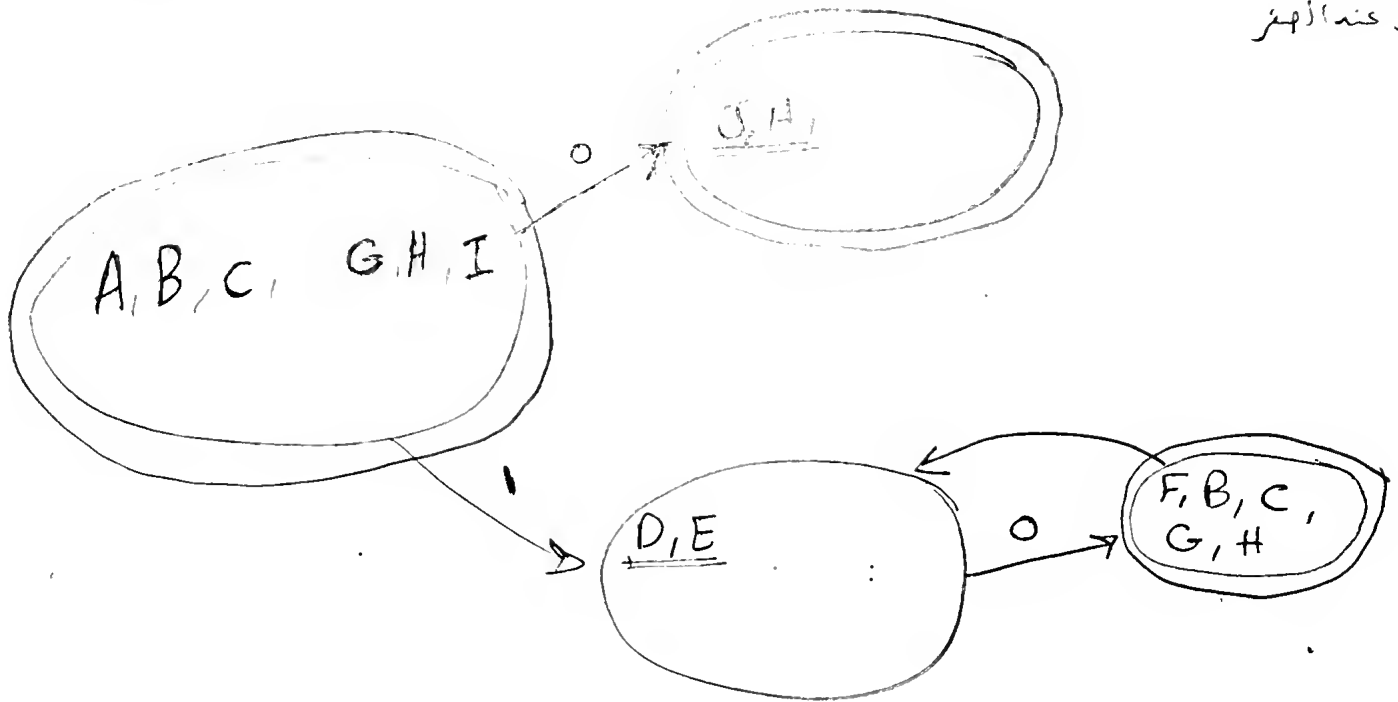




DFA

$A = \{ A, B, C, \dots, G, H, I \}$

ج عندا ٻهر



→ To implement DFA you may use table.

مثلاً في آخر مرة ممكن

Columns (input choices)

rows  
(state)

	0	1
A	B	C
B		
C	D	
D		C

ممكن لو عاين كتملاً الفراغ  
دي هتكتب (state)  
لو حدها بحيث لو عندك حالة  
ممكن تتش لاي حالة اخرى  
تكتب ال (state)  
دي

(optional) مش اجباري عليه

implementation  
i = 0 ; (index) for input char:  
state = 0 ; start state

while (input[i])

طالعني (i/p) متحرك زي  
i++

{

state = T[state, input[i]]

مفروض دول ينقلني  
next state

}

++i